CLAIMS

What is claimed is:

1. A method for handling a number of exceptions within a processor in a multi-processing system, the method comprising:

receiving an exception within the processor, wherein each processor in the multi-processor system shares a same memory;

executing a number of instructions at an address within a common interrupt handling vector address space of the same memory, wherein the number of instructions cause the processor to determine an identification of the processor based on a query that is internal to the processor; and

modifying execution flow of the exception to execute an interrupt handler located within one of a number of different interrupt handling vector address spaces.

2. The method of claim 1, wherein each processor in the multi-processor system executes one of a number of operating systems.

3. The method of claim 2, wherein each of the number of operating systems is associated with one of the number of different interrupt handling vector address spaces.

4. The method of claim 1, wherein the query that is internal to the processor includes reading a bit within a register that internal to the processor.

5. The method of claim 4, wherein the register is not dedicated to determining the identification of the processor.

6. The method of claim 1, further comprising determining the identification of the processor during initialization of the processor, based communications with a memory controller that is coupled between the processors in the multiple processor system and the same memory.

7. The method of claim 1, wherein each of the number of exceptions received by the different processors execute instructions at an address within the common interrupt handling vector address space of the same memory.

8. A method comprising:

receiving an exception within a processor, wherein the processor is included in a multi-processor system, wherein each processor in the multi-processor system executes one of a number of operating systems, wherein each processor in the multi-processor system shares a same memory and wherein the same memory includes a common interrupt handling address space and a number of different interrupt handling address spaces associated with each of the different processors in the multi-processor system;

determining the type of exception received within the processor;

executing a number of instructions at an address within the common interrupt handling address space of the same memory, wherein the number of instructions cause the processor to read a bit within an internal register to determine an identification of the processor in the multi-processor system; and

modifying execution flow of the exception to execute an interrupt handler located within one of the number of different interrupt handling address spaces.

9. The method of claim 8, wherein the internal register is not dedicated to determining the identification of the processor.

10. The method of claim 8, wherein each of the number of operating systems is associated with one of the number of different interrupt handling address spaces.

11. The method of claim 8, further comprising determining the identification of the processor during initialization of the processor, based communications with a memory controller that is coupled between the processors in the multiple processor system and the same memory.

12. A system comprising:

a memory that includes:

a common exception handling vector address space; and

a number of exception handling vector address spaces;

a memory controller coupled to the memory;

a first processor coupled to the memory controller, wherein the first processor is to execute a first operating system; and

a second processor coupled to the memory controller, wherein the second processor is to execute a second operating system, the second processor to execute a number of instructions in the common exception handling vector address space upon receipt of an exception, wherein the number of instructions cause the second processor to determine an identification of the second processor based on a query that is internal to the second processor, wherein the first processor and the second processor share the memory.

13. The system of claim 12, the first operating system and the second operating system are each associated with one of the number of different interrupt handling vector address spaces.

14. The system of claim 12, wherein the second processor further comprises an internal register and wherein the query that is internal to the second processor includes reading a bit within the internal register.

15. The system of claim 14, wherein the internal register is not dedicated to determining the identification of the second processor.

16. The system of claim 12, wherein the second processor is to determine the identification of the second processor during initialization, based communications with the memory controller.

17. A system comprising:

a memory that includes:

a common exception handling vector address space; and

a number of exception handling vector address spaces;

a memory controller coupled to the memory;

a first processor coupled to the memory controller, wherein the first processor is to execute a first operating system; and

a second processor coupled to the memory controller, the second processor to include an internal register, wherein the second processor is to execute a second operating system, the second processor to execute a number of instructions in the common exception handling vector address space upon receipt of an exception, wherein the number of instructions cause the second processor to determine an identification of the second processor based on a value stored in the internal register, wherein the first processor and the second processor share the memory.

18. The system of claim 17, the first operating system and the second operating system are each associated with one of the number of different interrupt handling vector address spaces.

19. The system of claim 17, wherein the internal register is not dedicated to determining the identification of the second processor.

20. The system of claim 17, wherein the second processor is to determine the identification of the second processor during initialization, based communications with the memory controller.

21. A machine-readable medium that provides instructions for handling a number of exceptions within a processor in a multi-processing system, which when executed by a machine, causes the machine to perform operations comprising:

receiving an exception within the processor, wherein each processor in the multi-processor system shares a same memory;

executing a number of instructions at an address within a common interrupt handling vector address space of the same memory, wherein the number of instructions cause the processor to determine an identification of the processor based on a query that is internal to the processor; and

modifying execution flow of the exception to execute an interrupt handler located within one of a number of different interrupt handling vector address spaces.

22. The machine-readable medium of claim 21, wherein each processor in the multi-processor system executes one of a number of operating systems.

23. The machine-readable medium of claim 22, wherein each of the number of operating systems is associated with one of the number of different interrupt handling vector address spaces.

24. The machine-readable medium of claim 21, wherein the query that is internal to the processor includes reading a bit within a register that internal to the processor.

25. The machine-readable medium of claim 24, wherein the register is not dedicated to determining the identification of the processor.

26. The machine-readable medium of claim 21, further comprising determining the identification of the processor during initialization of the processor, based communications with a memory controller that is coupled between the processors in the multiple processor system and the same memory.

27. The machine-readable medium of claim 21, wherein each of the number of exceptions received by the different processors execute instructions at an address within the common interrupt handling vector address space of the same memory.

28. A machine-readable medium that provides instructions, which when executed by a machine, causes the machine to perform operations comprising:

receiving an exception within a processor, wherein the processor is included in a multi-processor system, wherein each processor in the multi-processor system executes one of a number of operating systems, wherein each processor in the multi-processor system shares a same memory and wherein the same memory includes a common interrupt handling address space and a number of different interrupt handling address spaces associated with each of the different processors in the multi-processor system;

determining the type of exception received within the processor;

executing a number of instructions at an address within the common interrupt handling address space of the same memory, wherein the number of instructions cause the processor to read a bit within an internal register to determine an identification of the processor in the multi-processor system; and

modifying execution flow of the exception to execute an interrupt handler located within one of the number of different interrupt handling address spaces.

29. The machine-readable medium of claim 28, wherein the internal register is not dedicated to determining the identification of the processor.

30. The machine-readable medium of claim 28, wherein each of the number of operating systems is associated with one of the number of different interrupt handling address spaces.

31. The machine-readable medium of claim 28, further comprising determining the identification of the processor during initialization of the processor, based communications with a memory controller that is coupled between the processors in the multiple processor system and the same memory.